

PROGETTO S2L5

1. Capire cosa fa il programma senza eseguirlo.

Analizzando il codice del programma possiamo notare come è strutturato.

Una volta ricevuta la frase di benvenuto, possiamo decidere come proseguire, se:

- Moltiplicare due numeri
- Dividere due numeri
- Inserire una stringa

Se decidiamo di moltiplicare due numeri, avremo la possibilità di inserire in input due numeri e successivamente ci stamperà a schermo i due numeri inseriti e il prodotto.

Se decidiamo di dividere due numeri, inseriremo i due numeri e poi ci restituirà i due numeri inseriti e il risultato della divisione.

Con la terza opzione, il programma ci permetterà di inserire una stringa.

2. Individuare dal codice sorgente le casistiche non standard che il programma non gestisce.

Una casistica di un comportamento che non è stata utilizzata è quella dell'inserimento della stringa. Il codice al momento ci permette solamente di inserire la parola, ma non procede con un utilizzo della stessa. Possiamo utilizzarla per dare in output la seconda lettera della stringa, in questa maniera:

```
void ins_string ()
{
    char stringa[10];
    printf ("Inserisci la stringa:");
    scanf ("%s", &stringa);
    printf ("La stringa inserita e': %c", stringa[1]);
}
```

Un'altra cosa che ho notato è la mancanza di un controllore negli input del codice, portando quindi all'impossibilità di proseguire per un errore di digitazione o quando i caratteri utilizzati sono in minuscolo. Si può implementare una soluzione del tipo:

```
switch (scelta)
{
    case 'A':
        moltiplica();
        break;
    case 'B':
```

```

        dividi();

        break;

        case 'C':

            ins_string();

            break;

    case 'D':

        uscita();

        break;

    default :

        printf ("Risposta non prevista. Riprova con (A) (B) (C) (D)\n");

    }

```

Anche la possibilità di tornare al menu principale una volta utilizzata una funzione dello stesso non è stata prevista, così come un'opzione di uscita. Per questo motivo ho preferito inserire un ritorno al menu principale una volta esaurito l'utilizzo di una funzione. In più ho aggiunto una quarta opzione del menu con la possibilità di uscire dal programma e terminarlo.

3. Individuare eventuali errori di sintassi / logici e proporre una soluzione ad ognuno di essi

Un primo errore è

```

char scelta = {'\0'};

menu ();

scanf ("%d", &scelta)

```

in quanto la D preleva un numero e non una lettera, come indicato nel messaggio.

Per risolverlo basta cambiarlo in questo modo → `scanf ("%c", &scelta);`

Altro errore è nella sezione `void moltiplica()`, sempre nello `scanf ("%f", &a);scanf ("%d", &b);` poiché non preleva in input un numero. Possiamo risolverlo sempre con `scanf ("%hd", &a); scanf ("%hd", &b);` useremo `%hd` in quanto abbiamo dichiarato i due moltiplicatori come "short int".

Altro errore è in `int divisione = a % b;` qui notiamo un errore di operatore, in quanto `%` riporta il resto della divisione, ma non la effettua. In più l'attuale dichiarazione delle variabili non permette un risultato con la virgola. Per questo motivo un'ottima risoluzione al problema di questa parte del codice è:

```

void dividi ()

```

```

{
    int a,b = 0;

    printf ("Inserisci il numeratore:");
    scanf ("%d", &a);

    printf ("Inserisci il denominatore:");
    scanf ("%d", &b);

    float divisione = (float)a / b;

    printf ("La divisione tra %d e %d e': %.2f", a,b,divisione);
}

```

Anche se è una funziona praticamente non utilizzata dal programma, è bene notare che la variabile stringa è stata dichiarata come un'entità di 10 caratteri. Questo comporta ad un errore nel caso in cui inserissimo una stringa maggiore.

```

char stringa[10];

printf ("Inserisci la stringa:");
scanf ("%s", &stringa);

printf ("La stringa inserita e': %s", stringa)

```

una possibile soluzione sarebbe quella di aumentare la disponibilità dei caratteri → char stringa [20].

Per completezza e migliore leggibilità inserisco qui di seguito il codice corretto.

```
#include <stdio.h>
```

```

void menu ();
void moltiplica ();
void dividi ();
void ins_string();
void uscita();

```

```
int main ()
```

```

{
    char scelta = {'\0'};

```

```

while (scelta != 'D') {
    menu();
    scanf ("%c", &scelta);

    switch (scelta)
    {
        case 'A':
            moltiplica();
            break;
        case 'B':
            dividi();
            break;
        case 'C':
            ins_string();
            break;
        case 'D':
            uscita();
            break;
        default :
            printf ("Risposta non prevista. Riprova con (A) (B) (C) (D)\n");
    }
}
return 0;
}

```

```

void menu ()
{
    printf ("Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti\n");
    printf ("Come posso aiutarti?\n");
    printf ("A >> Moltiplicare due numeri\nB >> Dividere due numeri\nC >> Inserire una stringa
per prendere la seconda lettera\nD >> Esci dal gioco\n");
}

void moltiplica ()
{
    short int a,b = 0;
    printf ("Inserisci i due numeri da moltiplicare:\n");
    scanf ("%hd", &a);
    scanf ("%hd", &b);

    short int prodotto = a * b;

    printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto);
    printf ("\n");
}

```

```

void dividi ()
{
    int a,b = 0;
    printf ("Inserisci il numeratore:");
    scanf ("%d", &a);
    printf ("Inserisci il denominatore:");
    scanf ("%d", &b);

    float divisione = (float)a / b;

    printf ("La divisione tra %d e %d e': %.2f", a,b,divisione);
    printf ("\n");
}
void ins_string ()
{
    char stringa[20];
    printf ("Inserisci la stringa:");
    scanf ("%s", &stringa);
    printf ("La stringa inserita e': %c", stringa[1]);
    printf ("\n");
}
void uscita()
{printf("Sei uscito dal gioco\n");
}

```